Package ‘streamDepletr’

October 14, 2022

Title Estimate Streamflow Depletion Due to Groundwater Pumping

Version 0.1.1

Description Implementation of analytical models for estimating streamflow depletion due to groundwater pumping, and other related tools. Functions are broadly split into two groups: (1) analytical streamflow depletion models, which estimate streamflow depletion for a single stream reach resulting from groundwater pumping; and (2) depletion apportionment equations, which distribute estimated streamflow depletion among multiple stream reaches within a stream network. See Zipper et al. (2018) <doi:10.1029/2018WR022707> for more information on depletion apportionment equations and Zipper et al. (2019) <doi:10.1029/2018WR024403> for more information on analytical depletion functions, which combine analytical models and depletion apportionment equations.

Maintainer Samuel C. Zipper <samzipper@ku.edu>

Copyright Foundry Spatial Ltd.

URL https://github.com/FoundrySpatial/streamDepletr

BugReports https://github.com/FoundrySpatial/streamDepletr/issues

Depends R (>= 3.5)

License BSD_3_clause + file LICENSE

Imports Rmpfr, dplyr, magrittr, sp

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests knitr, rmarkdown, dismo, raster, deldir, rgeos, ggplot2, reshape2, scales

VignetteBuilder knitr

NeedsCompilation no

Author Samuel C. Zipper [aut, cre] (<https://orcid.org/0000-0002-8735-5757>), Ben Kerr [ctb], Tom Gleeson [ctb] (<https://orcid.org/0000-0001-9493-7707>), Qiang Li [rev] (<https://orcid.org/0000-0003-2221-6905>), Foundry Spatial Ltd. [cph]
**apportion_inverse**

Distribute streamflow depletion within a stream network using inverse distance weighting.

**Description**

Distribute streamflow depletion within a stream network using inverse distance weighting.

**Usage**

```r
apportion_inverse(
  reach_dist,
  w,
  max_dist = Inf,
  min_frac = 0,
  reach_name = NULL,
  dist_name = NULL
)
```
Arguments

reach_dist  data frame with two columns: reach, which is a grouping variable with the name of each stream reach, and dist which is the distance of that stream reach to the well of interest. There can be more than one dist per reach; the function will automatically find the minimum. Columns can either be named exactly as defined here, or set using reach_name and dist_name.

w  inverse distance weighting factor: 1 for inverse distance, 2 for inverse distance squared.

max_dist  the maximum distance of a stream to be depleted; defaults to Inf, which means all reaches will be considered.

min_frac  the minimum frac_depletion to be returned; defaults to 0, which means all reaches will be considered. If min_frac > 0 and some reaches have an estimated frac_depletion < min_frac, depletion in those reaches will be set to 0 and that depletion will be reallocated based on the proportional depletion in the remaining reaches.

reach_name  The name of the column in reach_dist indicating your stream reach grouping variable. If set to NULL (default), it will assume that the column name is reach.

dist_name  The name of the column in reach_dist indicating your distance variable. If set to NULL (default), it will assume that the column name is dist.

Details

Since analytical models assume the presence of 1 (or sometimes 2) linear streams, the `apportion_inverse` functions can be used to distribute that depletion to various reaches within a real stream network. These geometric functions are described in Zipper et al (2018), which found that `apportion_web` a weighting factor (w) of 2 produced the best results.

Value

A data frame with two columns:

reach  the grouping variable input in reach_dist
frac_depletion  the proportion of streamflow depletion from the well occurring in that reach.

References


Examples

reach_dist <- data.frame(reach = seq(1,5), dist = c(100, 150, 900, 300, 200))
apportion_inverse(reach_dist, w = 2)
apportion_inverse(reach_dist, w = 2, max_dist = 500)

reach_dist <- data.frame(reach = c("A", "A", "A", "B", "B"), dist = c(100, 150, 900, 300, 200))
apportion_inverse(reach_dist, w = 1)
apportion_inverse(reach_dist, w = 1, max_dist = 500)
apportion_polygon  

Distribute streamflow depletion within a stream network using web distance Thiessen polygons.

Description

Distribute streamflow depletion within a stream network using web distance Thiessen polygons.

Usage

apportion_polygon(
  reach_dist_lon_lat,
  wel_lon,
  wel_lat,
  crs,
  max_dist = Inf,
  min_frac = 0,
  reach_name = NULL,
  dist_name = NULL,
  lon_name = NULL,
  lat_name = NULL
)

Arguments

reach_dist_lon_lat  
data frame with four columns: reach, which is a grouping variable with the name of each stream reach; dist which is the distance of a point on that stream reach to the well of interest; lon which is the longitude of that point on the stream; and lat which is the latitude of that point on the stream. There can (and likely will) be multiple rows per reach. Columns can either be named exactly as defined here, or set using reach_name, dist_name, lon_name, and lat_name.

wel_lon  
longitude of well

wel_lat  
latitude of well

crs  
object of class CRS with projection info of latitude and longitude input

max_dist  
the maximum distance of a stream to be depleted; defaults to Inf, which means all reaches will be considered.

min_frac  
the minimum frac_depletion to be returned; defaults to 0, which means all reaches will be considered. If min_frac > 0 and some reaches have an estimated frac_depletion < min_frac, depletion in those reaches will be set to 0 and that depletion will be reallocated based on the proportional depletion in the remaining reaches.

reach_name  
The name of the column in reach_dist indicating your stream reach grouping variable. If set to NULL (default), it will assume that the column name is reach.
**apportion_web**

- **dist_name**
  The name of the column in `reach_dist` indicating your distance variable. If set to `NULL` (default), it will assume that the column name is `dist`.

- **lon_name**
  The name of the column in `reach_dist` indicating your longitude variable. If set to `NULL` (default), it will assume that the column name is `lon`.

- **lat_name**
  The name of the column in `reach_dist` indicating your latitude variable. If set to `NULL` (default), it will assume that the column name is `lat`.

**Details**

Since analytical models assume the presence of 1 (or sometimes 2) linear streams, the `apportion_*` functions can be used to distribute that depletion to various reaches within a real stream network. These geometric functions are described in Zipper et al (2018), which found that `apportion_web` a weighting factor \((w)\) of 2 produced the best results.

**Value**

A data frame with two columns:

- **reach** the grouping variable input in `reach_dist`
- **frac_depletion** the proportion of streamflow depletion from the well occurring in that reach.

**References**


**Examples**

```r
rdll <- prep_reach_dist(wel_lon = 295500, wel_lat = 4783200, 
                        stream_shp = stream_lines, reach_id = "reach", stream_pt_spacing = 1)
apportion_polygon(reach_dist_lon_lat = rdll, wel_lon = 295500, wel_lat = 4783200, 
                  max_dist = 5000, crs = raster::crs(stream_lines))
```

**Description**

Distribute streamflow depletion within a stream network using web distance weighting.
Usage

```r
apportion_web(
  reach_dist,
  w,
  max_dist = Inf,
  min_frac = 0,
  reach_name = NULL,
  dist_name = NULL
)
```

Arguments

- `reach_dist` data frame with two columns: `reach`, which is a grouping variable with the name of each stream reach, and `dist` which is the distance of a point on that stream reach to the well of interest. There can (and likely will) be more than one `dist` per `reach`; if there is only one `dist` per `reach`, results will be the same as the `apportion_inverse` method. Columns can either be named exactly as defined here, or set using `reach_name` and `dist_name`.

- `w` weighting factor; 1 for web, 2 for web squared.

- `max_dist` the maximum distance of a stream to be depleted; defaults to `Inf`, which means all reaches will be considered.

- `min_frac` the minimum `frac_depletion` to be returned; defaults to 0, which means all reaches will be considered. If `min_frac > 0` and some reaches have an estimated `frac_depletion < min_frac`, depletion in those reaches will be set to 0 and that depletion will be reallocated based on the proportional depletion in the remaining reaches.

- `reach_name` The name of the column in `reach_dist` indicating your stream reach grouping variable. If set to `NULL` (default), it will assume that the column name is `reach`.

- `dist_name` The name of the column in `reach_dist` indicating your distance variable. If set to `NULL` (default), it will assume that the column name is `dist`.

Details

Since analytical models assume the presence of 1 (or sometimes 2) linear streams, the `apportion_*` functions can be used to distribute that depletion to various reaches within a real stream network. These geometric functions are described in Zipper et al (2018), which found that `apportion_web` a weighting factor (`w`) of 2 produced the best results.

Value

A data frame with two columns:

- `reach` the grouping variable input in `reach_dist`
- `frac_depletion` the proportion of streamflow depletion from the well occurring in that reach.
References


Examples

```r
reach_dist <- data.frame(reach = seq(1,5),
                         dist = c(100, 150, 900, 300, 200))
apportion_web(reach_dist, w = 2) # same as inverse because there's only one dist per reach
apportion_web(reach_dist, w = 2, max_dist = 500)

reach_dist <- data.frame(reach = c("A", "A", "A", "B", "B"),
                         dist = c(100, 150, 900, 300, 200))
apportion_web(reach_dist, w = 1)
apportion_web(reach_dist, w = 1, max_dist = 500)
```

---

**apportion_wedge**

*Distribute streamflow depletion between two streams in a wedge-shaped aquifer.*

**Description**

Distribute streamflow depletion between two streams in a wedge-shaped aquifer.

**Usage**

```r
apportion_wedge(angle_total, angle_well)
```

**Arguments**

- `angle_total` angle [radians] between the two streams.
- `angle_well` angle [radians] from the first (lower boundary) stream and the well.

**Details**

This function assumes that streams are two linear tributaries which meet at the origin. This function specifically corresponds to Equations 18 and 19 in Yeh et al. (2008).

**Value**

A numeric of length two with the proportion of steady-state capture fraction from the first (lower) and second (upper) streams.

**References**

depletion_max_distance

Calculate maximum distance at which streamflow depletion will exceed a user-selected threshold. Note that this only considers a single stream - depletion apportionment does not occur.

Description

Calculate maximum distance at which streamflow depletion will exceed a user-selected threshold. Note that this only considers a single stream - depletion apportionment does not occur.

Usage

depletion_max_distance(
  Qf_thres = 0.01,
  d_interval = 100,
  d_min = NULL,
  d_max = 5000,
  method = "glover",
  t,
  S,
  Tr,
  ...
)

Arguments

Qf_thres streamflow depletion fraction (Qf) threshold used to define maximum distance. Defaults to 0.01 (1%).
d_interval interval to use for testing; this defines the spatial resolution at which output will be returned [L]
d_min minimum search distance [L]. If 'Qf' < 'Qf_thres' at 'd_min', function will return 'd_min' and a warning. If 'd_min'=NULL (default), 'd_min' will be set to 'd_interval'
d_max maximum search distance [L]. If 'Qf' > 'Qf_thres' at 'd_max', function will return 'd_max' and a warning.
method analytical solution to use (options= 'glover', 'hunt'). Defaults to 'glover'.
t time you want output for [T]
S aquifer storage coefficient (specific yield if unconfined; storativity if confined)
Tr aquifer transmissivity [L2/T]
... any other inputs required for your method of choice; for example, hunt needs lmda (streambed conductance)
Details

This function is useful for the 'Expanding' stream proximity criteria described in Zipper et al. (2018).

Value

A numeric of the distance at which streamflow depletion fraction ($Q_f$) drops below the threshold at time 't'.

References


Examples

depletion_max_distance(method = "glover", t = 730, S = 0.1, Tr = 100)
depletion_max_distance(Qf_thres = 0.001, method = "glover", t = 730, S = 0.1, Tr = 100)
depletion_max_distance(Qf_thres = 0.001, method = "hunt", t = 730, S = 0.1, Tr = 100, lmda = 0.01)
depletion_max_distance(Qf_thres = 0.001, method = "hunt", t = 7300, S = 0.1, Tr = 100, lmda = 0.01)

Description

Daily discharge data for the 2014-2015 water years for the US Geological Survey gauging stations DORN (SPRING) CREEK AT CT HIGHWAY M NR WAUNAKEE,WI (station ID 05427930) and SIXMILE CREEK @ COUNTY TRNK HGHWY M NR WAUNAKEE,WI (station ID 05427910).

Usage

discharge_df

Format

A data frame with 1450 rows and 2 variables:

- **date** date of streamflow measurement
- **Q_m3d** discharge, in cubic meters per day
- **stream** name of stream for each stream reach (Sixmile Creek or Dorn Creek) ...

Source

**glover**

*Streamflow depletion with fully-penetrating stream and no streambed.*

**Description**
Streamflow depletion with fully-penetrating stream and no streambed.

**Usage**
glover(t, d, S, Tr)

**Arguments**
- `t`: times you want output for [T]
- `d`: distance from well to stream [L]
- `S`: aquifer storage coefficient (specific yield if unconfined; storativity if confined)
- `Tr`: aquifer transmissivity [L^2/T]

**Details**
This function is described in Glover & Balmer (1954) based on work by Theis (1941). It contains numerous assumptions:

- Horizontal flow » vertical flow (Dupuit assumptions hold)
- Homogeneous, isotropic aquifer
- Constant `Tr`: Aquifer is confined, or if unconfined change in head is small relative to aquifer thickness
- Stream is straight, infinitely long, and remains in hydraulic connection to aquifer
- Constant stream stage
- No changes in recharge due to pumping
- No streambank storage
- Constant pumping rate
- Aquifer extends to infinity
- Stream fully penetrates through aquifer (see hunt or hantush for partially penetrating stream)
- No streambed resistance to flow (see hunt or hantush for streambed resistance)

**Value**
A numeric of \( Q_f \), streamflow depletion as fraction of pumping rate [-]. If the pumping rate of the well \( (Q_w; [L^3/T]) \) is known, you can calculate volumetric streamflow depletion \([L^3/T]\) as \( Q_f \times Q_w \)
References


Examples

glover(t = 1.5777e8, d = 1000, S = 0.2, Tr = 0.1) # Glover & Balmer (1954) Table 1, Well 1
glover(t = 1.5777e8, d = 5000, S = 0.2, Tr = 0.1) # Glover & Balmer (1954) Table 1, Well 2
glover(t = 1.5777e8, d = 10000, S = 0.2, Tr = 0.1) # Glover & Balmer (1954) Table 1, Well 3

hantush

Streamflow depletion in partially penetrating stream with semipervious streambed.

Description

Streamflow depletion in partially penetrating stream with semipervious streambed.

Usage

hantush(t, d, S, Kh, b, Kriv, briv, prec = 80)

Arguments

t | times you want output for [T]
d | distance from well to stream [L]
S | aquifer storage coefficient (specific yield if unconfined; storativity if confined)
Kh | aquifer horizontal hydraulic conductivity [L/T]
b | aquifer saturated thickness [L]
Kriv | streambed semipervious layer hydraulic conductivity [L/T]
briv | streambed semipervious layer thickness [L]
prec | precision for mpfr package for storing huge numbers; 80 seems to generally work but tweak this if you get weird results.

Details

This function is described in Hantush (1965). As the leakance term \((b*Kh/Kriv)\) approaches 0 this is equivalent to glover. It contains numerous assumptions:

- Horizontal flow » vertical flow (Dupuit assumptions hold)
- Homogeneous, isotropic aquifer
- Constant \(Tr\): Aquifer is confined, or if unconfined change in head is small relative to aquifer thickness
- Stream is straight, infinitely long, and remains in hydraulic connection to aquifer
- Constant stream stage
- No changes in recharge due to pumping
- No streambank storage
- Constant pumping rate
- Aquifer extends to infinity

**Value**

A numeric of $Q_f$, streamflow depletion as fraction of pumping rate [-]. If the pumping rate of the well ($Q_w; [L^3/T]$) is known, you can calculate volumetric streamflow depletion $[L^3/T]$ as $Q_f \times Q_w$

**References**


**Examples**

```r
hantush(t = 1826, d = 1000, S = 0.2, Kh = 86.4, b = 100, Kriv = 0.0864, briv = 1)
Qf <- hantush(t = seq(1, 1826), d = 1000, S = 0.2, Kh = 86.4, b = 100, Kriv = 0.0864, briv = 1)
plot(x = seq(1, 1826), y = Qf, type = "l")
```

**Description**

Streamflow depletion in partially penetrating stream with semipervious streambed.

**Usage**

```r
hunt(t, d, S, Tr, lmda, lmda_max = Inf, prec = 80)
```

**Arguments**

- `t` times you want output for [T]
- `d` distance from well to stream [L]
- `S` aquifer storage coefficient (specific yield if unconfined; storativity if confined)
- `Tr` aquifer transmissivity $[L^2/T]$
- `lmda` streambed conductance term, lambda $[L/T]$. Can be estimated with `streambed_conductance`.
- `lmda_max` maximum allowed 'lmda' $[L/T]$. If 'lmda' is too high, exp and erfc calculations in Hunt solution are not computationally possible, so you may need to artificially reduce 'lmda' using this term.
precision for Rmpfr package for storing huge numbers; 80 seems to generally work but tweak this if you get weird results. Reducing this value will reduce accuracy but speed up computation time.

Details

This function is described in Hunt (1999). When $\lambda$ term gets very large, this is equivalent to Glover. It contains numerous assumptions:

- Horizontal flow $\gg$ vertical flow (Dupuit assumptions hold)
- Homogeneous, isotropic aquifer
- Constant $Tr$: Aquifer is confined, or if unconfined change in head is small relative to aquifer thickness
- Stream is straight, infinitely long, and remains in hydraulic connection to aquifer
- Constant stream stage
- No changes in recharge due to pumping
- No streambank storage
- Constant pumping rate
- Aquifer extends to infinity

Value

A numeric of $Q_f$, streamflow depletion as fraction of pumping rate [-]. If the pumping rate of the well ($Q_w; [L^3/T]$) is known, you can calculate volumetric streamflow depletion $[L^3/T]$ as $Q_f*Q_w$

References


Examples

```r
hunt(t = 1826, d = 1000, S = 0.2, Tr = 8640, lmda = 864) # ~equal to glover because lmda=Tr
hunt(t = 1826, d = 1000, S = 0.2, Tr = 8640, lmda = 0.864) # less depletion due to lower lmda
lmda <- streambed_conductance(w = 10, Kriv = 0.0864, briv = 1) # estimate lmda
hunt(t = 1826, d = 1000, S = 0.2, Tr = 8640, lmda = lmda)
Qf <- hunt(t = seq(1, 1826), d = 1000, S = 0.2, Tr = 8640, lmda = 0.864)
plot(x = seq(1, 1826), y = Qf, type = "l")
```
**induce_infiltration_rate**

*Calculate the pumping rate at which pumping will induce infiltration from stream.*

---

**Description**

Calculate the pumping rate at which pumping will induce infiltration from stream.

**Usage**

```
induce_infiltration_rate(d, Qa)
```

**Arguments**

- `d` distance from well to stream [L]
- `Qa` ambient groundwater inflow rate per unit length of stream [L^2/T]

**Details**

This calculates the critical pumping rate above which induced infiltration due to groundwater pumping will occur, based on the glover model of streamflow depletion. Derived in Wilson (1993) Eq. 5.

**Assumptions:**

- Groundwater flow is perpendicular to stream
- Horizontal flow >> vertical flow (Dupuit assumptions hold)
- Homogeneous, isotropic aquifer
- Constant Tr: Aquifer is confined, or if unconfined change in head is small relative to aquifer thickness
- Stream is straight, infinitely long, and remains in hydraulic connection to aquifer
- Constant stream stage
- No changes in recharge due to pumping
- No streambank storage
- Constant pumping rate
- Aquifer extends to infinity
- Stream fully penetrates through aquifer
- No streambed resistance to flow (see hunt or hantush for streambed resistance)

**Value**

A numeric of Qc, critical pumping rate above which induced infiltration due to groundwater pumping will occur [L^3/T].
References


Examples

```r
induce_infiltration_rate(d = 100, Qa = 10)
induce_infiltration_rate(d = 100, Qa = 50)
induce_infiltration_rate(d = 500, Qa = 50)
```

```r
induce_infiltration_time
Calculate the critical time at which stream transitions from gaining to losing.
```

Description

Calculate the critical time at which stream transitions from gaining to losing.

Usage

```r
induce_infiltration_time(d, S, Tr, Qa, Qw)
```

Arguments

- `d`: distance from well to stream [L]
- `S`: aquifer storage coefficient (specific yield if unconfined; storativity if confined)
- `Tr`: aquifer transmissivity [L^2/T]
- `Qa`: ambient groundwater inflow rate per unit length of stream [L^2/T]
- `Qw`: well pumping rate [L^3/T]

Details

This calculates the critical time at which induced infiltration due to groundwater pumping begins, based on the `glover` model of streamflow depletion. Derived in Chen (2003) Eq. 4.

Assumptions:

- Groundwater flow is perpendicular to stream
- Horizontal flow » vertical flow (Dupuit assumptions hold)
- Homogeneous, isotropic aquifer
- Constant `Tr`: Aquifer is confined, or if unconfined change in head is small relative to aquifer thickness
- Stream is straight, infinitely long, and remains in hydraulic connection to aquifer
- Constant stream stage
intermittent_pumping

- No changes in recharge due to pumping
- No streambank storage
- Constant pumping rate
- Aquifer extends to infinity
- Stream fully penetrates through aquifer
- No streambed resistance to flow (see hunt or hantush for streambed resistance)

Value

A numeric of tc, the critical time at which induced infiltration begins [T].

References


Examples

# recreate Figure 2 in Chen (2003)
Qa <- c(0.0001, 0.0003, 0.0005, 0.0008, 0.001)
tc <- induce_infiltration_time(d = 575, S = 0.2, Tr = 100*15, Qa = Qa, Qw = 2727)
plot(x = (pi * Qa * 100 * 15 * 575 / 2727), y = tc, log = "y")

intermittent_pumping

Streamflow depletion for an intermittent pumping schedule using superposition.

Description

Streamflow depletion for an intermittent pumping schedule using superposition.

Usage

intermittent_pumping(t, starts, stops, rates, method = "glover", d, S, Tr, ...)

Arguments

t times you want output for [T]
starts vector of times to start pumping [T] (must be same length as stops and rates)
stops vector of times pumping stops [T] (must be same length as starts and rates)
rates vector of pumping rates [L3/T] (must be same length as starts and stops)
method analytical solution to use (options= ‘glover’, ‘hunt’). Defaults to ‘glover’.
d distance from well to stream [L]
S aquifer storage coefficient (specific yield if unconfined; storativity if confined)
Tr aquifer transmissivity [L2/T]
... any other inputs required for your method of choice; for example, hunt needs 
Lmda (streambed conductance)
Details

This function superimposes wells and image wells to calculate a timeseries of streamflow depletion. Unlike the streamflow depletion models (e.g. glover, hunt) this is not fractional depletion (Qf) because there can be different pumping rates at different times.

Value

A numeric of Qs, streamflow depletion rate [L3/T].

References


Examples

Qs <- intermittent_pumping(t = seq(0, 1000, 5),
  starts = seq(0, 900, 10), stops = seq(9, 909, 10), rates = seq(1, 1000, length.out=91),
  method = “hunt”, d = 100, S = 0.1, Tr = 100, lmda = 10)

Qs <- intermittent_pumping(t = seq(0, 1000, 5),
  starts = seq(0, 900, 10), stops = seq(9, 909, 10), rates = seq(1, 1000, length.out=91),
  method = “hunt”, d = 100, S = 0.1, Tr = 100, lmda = 100000, lmda_max = 10)

Description

Calculate the distance from a well to each reach within a stream network. This function splits a polyline stream network up into a series of evenly spaced points and calculates the distance from each of those points to a well.

Usage

prep_reach_dist(wel_lon, wel_lat, stream_shp, reach_id, stream_pt_spacing, buffer_width = 0.1, nseed = 1)
streambed_conductance

Arguments

- `wel_lon`  longitude of well
- `wel_lat`  latitude of well
- `stream_shp`  shapefile of stream reaches
- `reach_id`  string indicating name of column in `stream_shp` that
- `stream_pt_spacing`  distance between points used for sampling each stream reach. The actual distance between points will be close to this (but not necessarily exact) due to sampling rounding error. The finer spacing you use, the more accurate your results will be but the function will run slower and use more memory.
- `buffer_width`  width of buffer around stream used to match points with polylines
- `nseed`  seed for random number generator (this is used to convert stream polylines to points)

Value

A data frame with four columns:

- **reach**  a grouping variable with the name of each stream reach
- **dist**  distance of a point on that stream reach to the well of interest
- **lat**  latitude of that point
- **lon**  longitude of that point

This data frame can be plugged directly into apportion_inverse, apportion_polygon (if `latlon=T`), or apportion_web

Examples

```r
rdll <- prep_reach_dist(wel_lon = 295500, wel_lat = 4783200, stream_shp = stream_lines, reach_id = "reach", stream_pt_spacing = 1)
head(rdll)
```

streambed_conductance  Estimate streambed conductance.

Description

Estimate streambed conductance.

Usage

```r
streambed_conductance(w, Kriv, briv)
```
Arguments

- \( w \) stream width [L]
- \( Kriv \) streambed semipervious layer hydraulic conductivity [L/T]. Reeves et al. (2009) estimate this as the vertical hydraulic conductivity of the aquifer (\( K_v \); L/T), which is itself often estimated as 10% of the horizontal hydraulic conductivity (\( K_h \times 0.1 \); L/T)
- \( briv \) streambed semipervious layer thickness [L] Reeves et al. (2009) estimate this as the vertical distance from the streambed to the top of the well screen, or the length of the well screen, whichever is greater [L].

Value

A numeric of \( \lambda \), the streambed conductance term [L/T]

References


Examples

- streambed_conductance(w = 10, Kriv = 0.0864, briv = 1)
- streambed_conductance(w = 5, Kriv = 0.0864, briv = 1)
- streambed_conductance(w = 10, Kriv = 0.864, briv = 1)
- streambed_conductance(w = 10, Kriv = 0.0864, briv = 0.1)

Stream network for Sixmile Creek Watershed, Wisconsin, USA. Extracted from US NHDPlus v2.1 national seamless dataset.

Description

Stream network for Sixmile Creek Watershed, Wisconsin, USA. Extracted from US NHDPlus v2.1 national seamless dataset.

Usage

stream_lines

Format

A SpatialLinesDataFrame with 49 rows and 2 variables:

- \( reach \) identifier code for each stream reach
- \( stream \) name of stream for each stream reach (Sixmile Creek or Dorn Creek) ...
Source

http://www.horizon-systems.com/NHDPlusData/NHDPlusV21/Data/NationalData/
Index

* datasets
  discharge_df, 9
  stream_lines, 19

apportion_inverse, 2, 6, 18
apportion_polygon, 4, 18
apportion_web, 5, 18
apportion_wedge, 7

depletion_max_distance, 8
discharge_df, 9

glover, 10, 11, 13–15, 17

hantush, 10, 11, 14, 16
hunt, 10, 12, 14, 16, 17

induce_infiltration_rate, 14
induce_infiltration_time, 15
intermittent_pumping, 16

pred_reach_dist, 17

stream_lines, 19
streambed_conductance, 18